PROJECT 2 - GENERATING DIGITAL AUDIO

Project Overview

In this project, you will create a C program that generates a sine wave—a fundamental audio signal—and saves it as raw PCM (Pulse Code Modulation) data in a file. The output will be a .raw file containing uncompressed audio samples, which can later be imported into audio software (e.g., Audacity) as PCM data for playback. This project simulates how digital audio is created and stored, offering hands-on experience with signal generation and file I/O in C.

Objectives

• Generate a sine wave with customizable frequency, amplitude, and duration.
• Encode the sine wave as PCM data using a specified sample rate and bit depth.
• Write the PCM data to a file in a raw format.
• Explore key C programming concepts through practical application.

Project Features

1. User Input: Accept parameters like frequency (e.g., 440 Hz for an A note), duration (in seconds), and amplitude (volume level).
2. Sine Wave Generation: Use the sin() function from the math library to compute sine wave samples.
3. PCM Encoding: Convert the sine wave into 16-bit PCM samples (signed integers).
4. File Output: Save the PCM data to a .raw file, which can be played back with proper audio tools.
5. Basic Error Handling: Check for file creation issues and invalid user inputs.

Implementation Steps

1. Setup: Include necessary headers (stdio.h, stdlib.h, math.h) and define constants (e.g., sample rate = 44,100 Hz, bit depth = 16).
2. Input Handling: Use scanf() to get frequency, duration, and amplitude from the user.
3. Sine Wave Calculation: Compute samples using the formula sample = amplitude * sin(2 * PI * frequency * t / sample_rate), where t is the sample index.
4. PCM Conversion: Scale the floating-point sine values (-1.0 to 1.0) to 16-bit integers (-32,768 to 32,767).
5. File Writing: Open a file in binary mode (fopen("output.raw", "wb")) and write the samples using fwrite().
6. Cleanup: Close the file and handle any errors.

Sample Output

• A 440 Hz sine wave for 5 seconds at 44.1 kHz sample rate will produce a 441,000-byte file (44,100 samples/sec × 5 sec × 2 bytes/sample for 16-bit mono).
• Import the .raw file into Audacity (File > Import > Raw Data, 44.1 kHz, 16-bit PCM, mono) to hear the tone.

C Language Features Learners Can Explore

1. Basic I/O Operations – Use printf() and scanf() for user interaction.
2. Mathematical Functions – Apply math.h and sin(); compile with -lm.
3. Data Types and Casting – Convert between double and int16_t.
4. Loops and Iteration – Generate samples using for loops.
5. File Handling – fopen(), fwrite(), fclose() for binary I/O.
6. Pointers and Memory Management – Pass data efficiently to fwrite().
7. Error Handling – Validate fopen() and fwrite() results.
8. Constants and Preprocessor Directives – Use #define for constants.
9. Arrays – Store samples using dynamic memory allocation.
10. Modularity – Organize code into functions.

Compile and Run

• Compile: gcc main.c -lm
• Run: ./a.out
• Play: Import in Audacity and play.

Learning Extensions

• Stereo Output: Write two-channel (left and right) data.
• Multiple Frequencies: Generate chords by summing multiple sine waves.
• WAV Header: Add a header for native playback.